David Keyes, project lead

http://www.tops-scidac.org

SciDAC
Scientific Discovery through Advanced Computing

Who we are…

… the PETSc and TAO people

… the hypre and Sundials people

Berkeley Lab

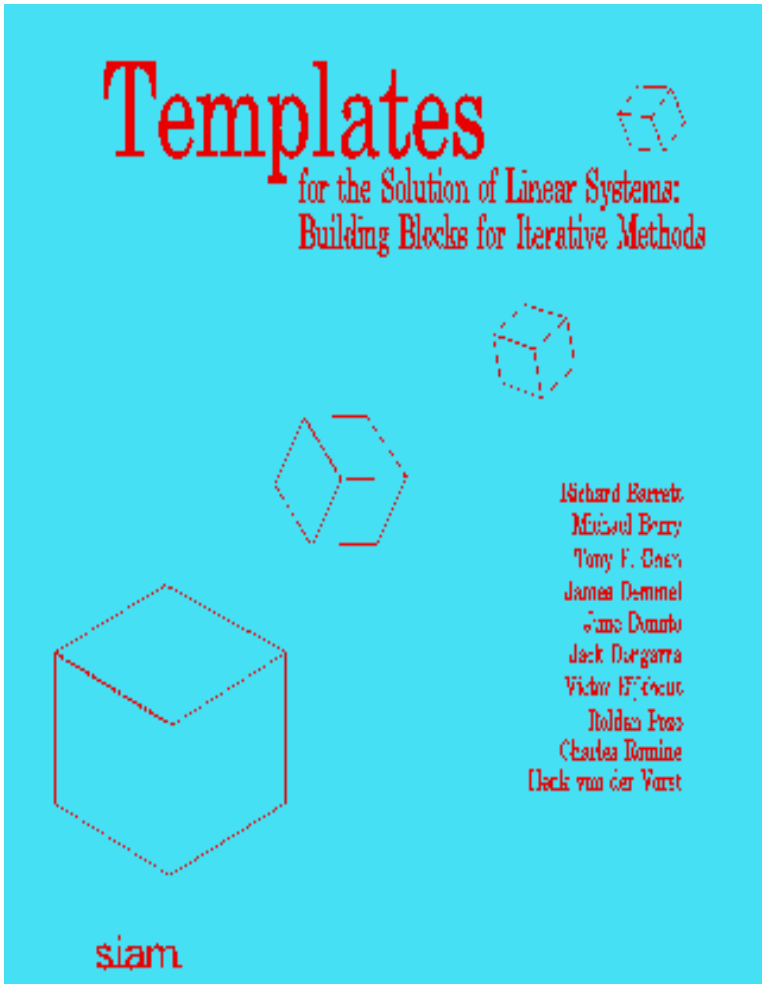
… the SuperLU and PARPACK people

Plus some university collaborators …

… with a history of lab collaborations in high performance computing

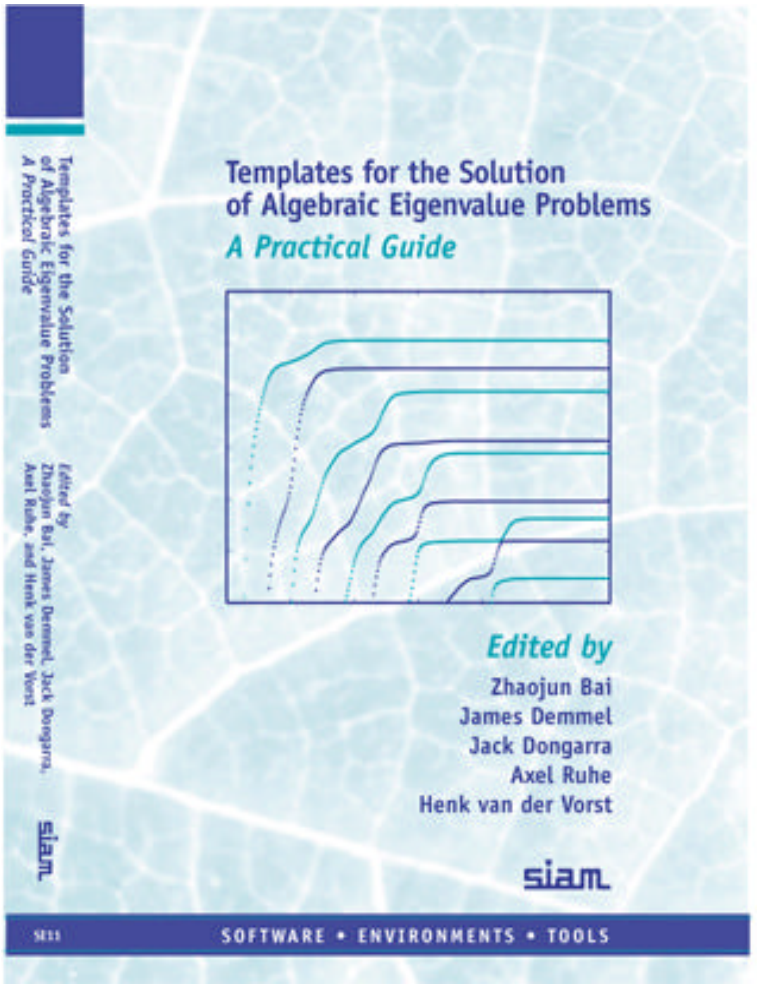
You may know the on-line "Templates" guides …

www.netlib.org/templates



124 pp.

www.netlib.org/etemplates

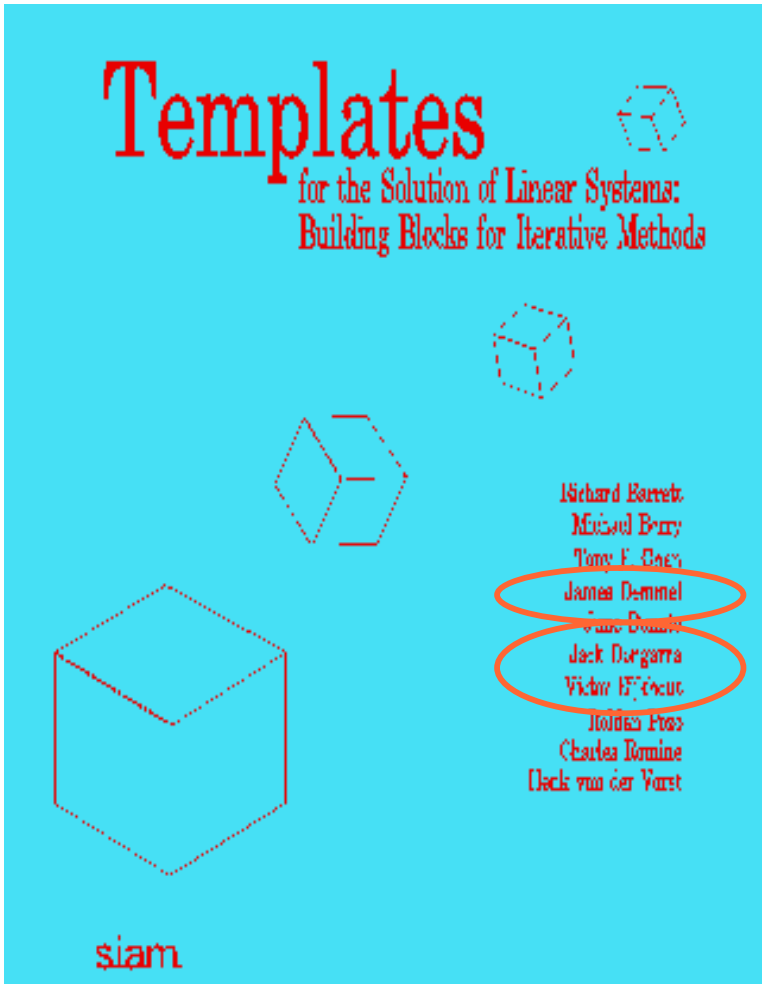


410 pp.

… these are good starts, but not adequate for SciDAC scales!

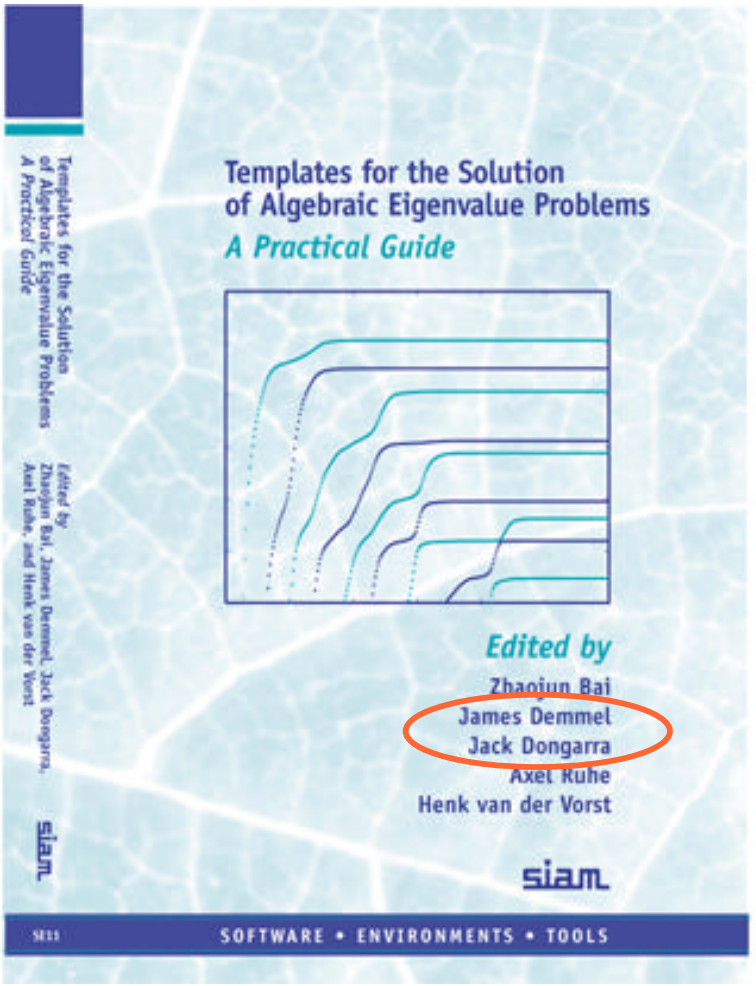
You may know the on-line "Templates" guides …

www.netlib.org/templates



124 pp.

www.netlib.org/etemplates

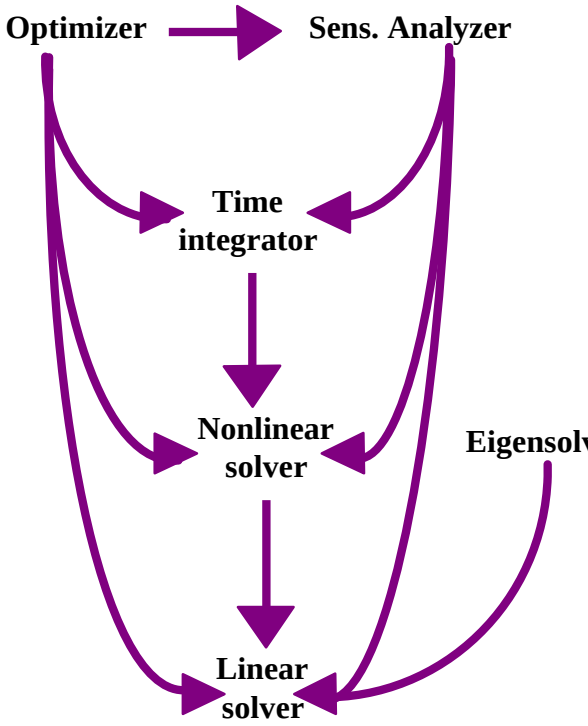


410 pp.

… SciDAC puts some of the authors (and many others) "on-line" for you

Scope for TOPS

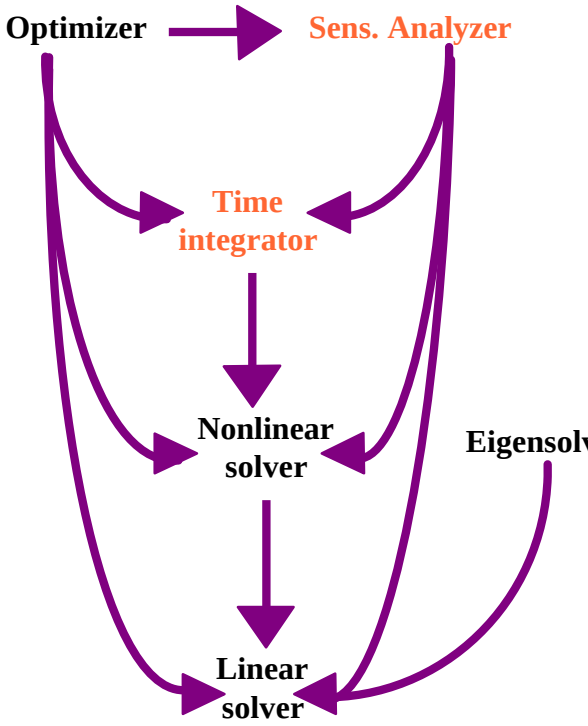
- **Design and implementation of "solvers"**
  - **Time integrators (w/ sens. anal.)** $f(\dot{x}, x, t, p) = 0$
  - **Nonlinear solvers (w/ sens. anal.)** $F(x, p) = 0$
  - **Optimizers** $\min_u f(x,u)\ s.t.\ F(x,u) = 0, u \geq 0$
  - **Linear solvers** $Ax = b$
  - **Eigensolvers** $Ax = lBx$



- **Software integration**
- **Performance optimization**

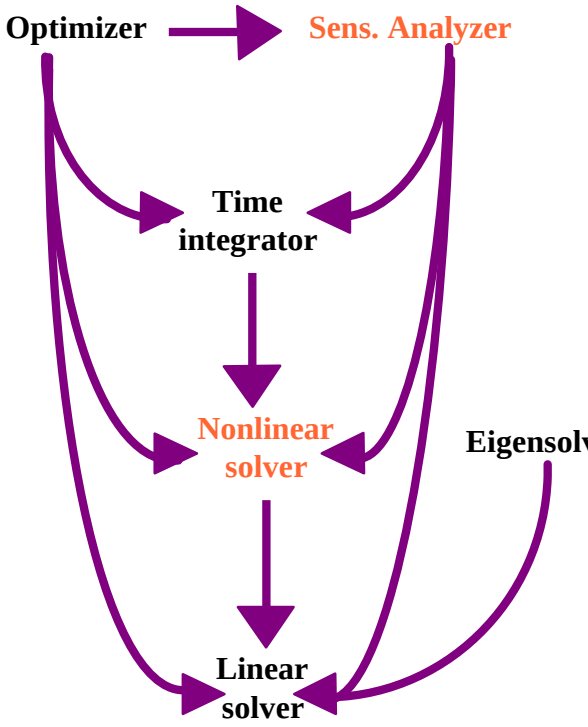
Scope for TOPS

- **Design and implementation of "solvers"**
  - **Time integrators (w/ sens. anal.)** $f(\dot{x}, x, t, p) = 0$
  - **Nonlinear solvers (w/ sens. anal.)** $F(x, p) = 0$
  - **Optimizers** $\min_u f(x,u)\ s.t.\ F(x,u) = 0, u \geq 0$
  - **Linear solvers** $Ax = b$
  - **Eigensolvers** $Ax = lBx$
- **Software integration**
- **Performance optimization**



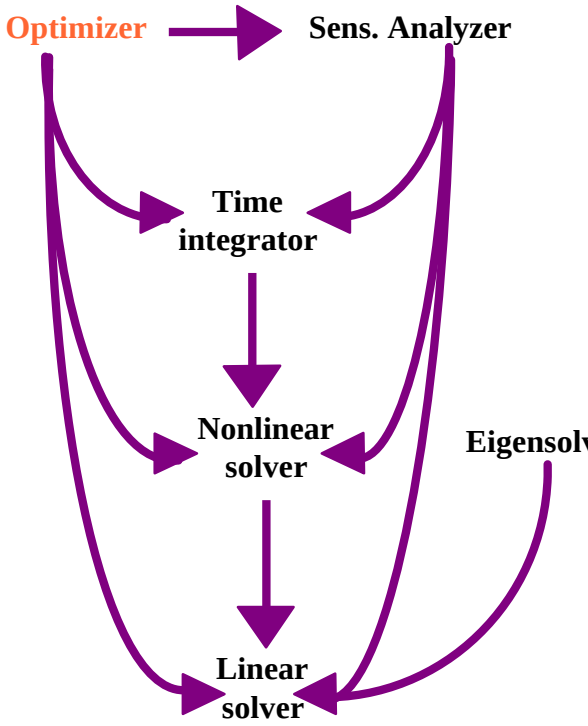
Scope for TOPS

- **Design and implementation of "solvers"**
  - **Time integrators (w/ sens. anal.)** $f(\dot{x}, x, t, p) = 0$
  - **Nonlinear solvers (w/ sens. anal.)** $F(x, p) = 0$
  - **Optimizers** $\min_u f(x,u)\ s.t.\ F(x,u) = 0, u \geq 0$
  - **Linear solvers** $Ax = b$
  - **Eigensolvers** $Ax = lBx$
- **Software integration**
- **Performance optimization**



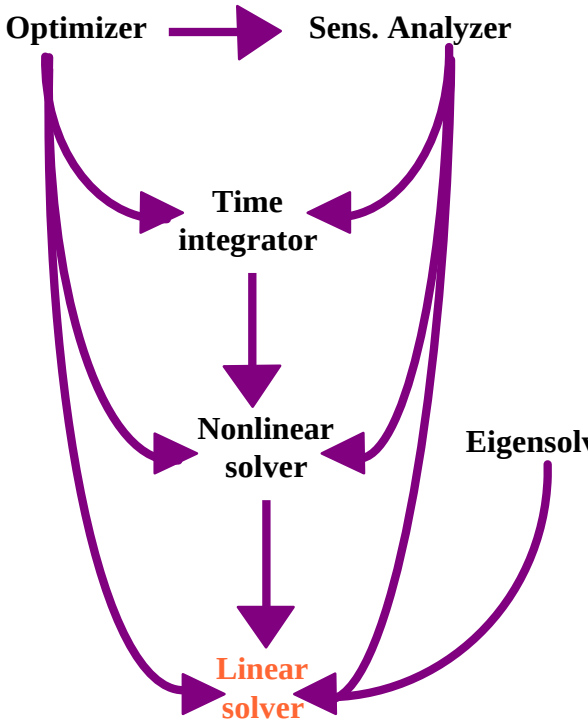
Scope for TOPS

- **Design and implementation of "solvers"**
  - **Time integrators (w/ sens. anal.)** $f(\dot{x}, x, t, p) = 0$
  - **Nonlinear solvers (w/ sens. anal.)** $F(x, p) = 0$
  - **Optimizers** $\min_u f(x,u)\ s.t.\ F(x,u) = 0, u \geq 0$
  - **Linear solvers** $Ax = b$
  - **Eigensolvers** $Ax = lBx$
- **Software integration**
- **Performance optimization**



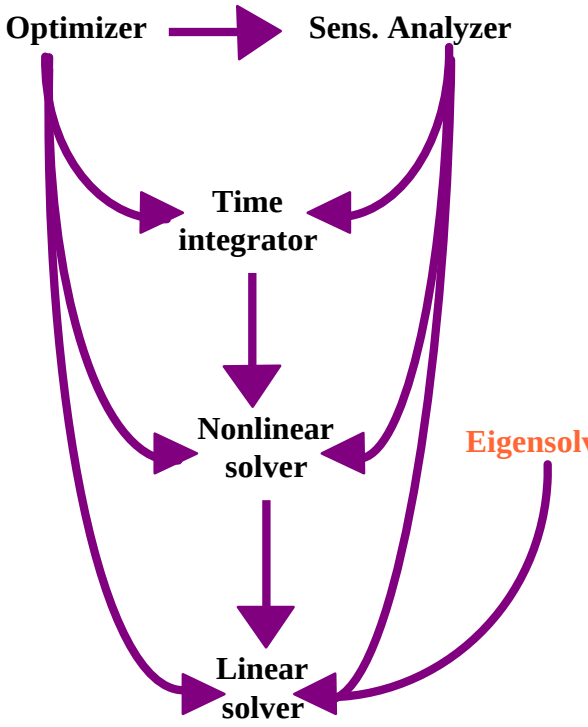
Scope for TOPS

- **Design and implementation of "solvers"**
  - **Time integrators (w/ sens. anal.)** $f(\dot{x}, x, t, p) = 0$
  - **Nonlinear solvers (w/ sens. anal.)** $F(x, p) = 0$
  - **Optimizers** $\min_u f(x,u)\ s.t.\ F(x,u) = 0, u \geq 0$
  - **Linear solvers** $Ax = b$
  - **Eigensolvers** $Ax = lBx$
- **Software integration**
- **Performance optimization**



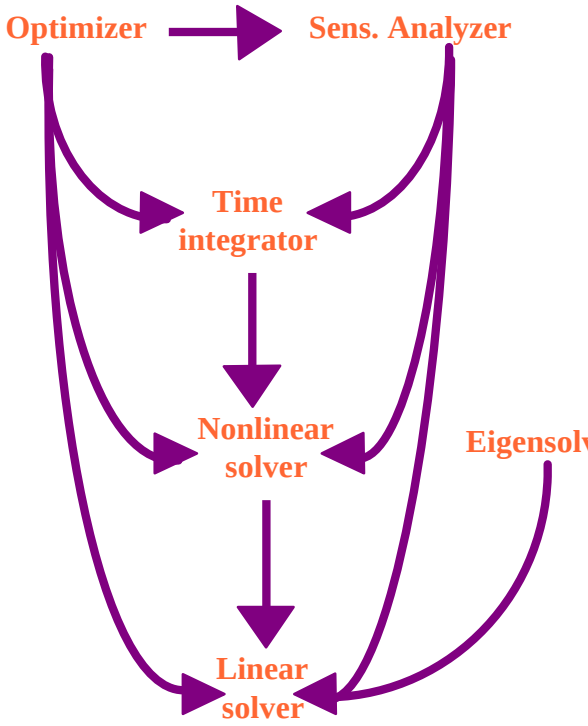
Scope for TOPS

- **Design and implementation of "solvers"**
  - **Time integrators (w/ sens. anal.)** $f(\dot{x}, x, t, p) = 0$
  - **Nonlinear solvers (w/ sens. anal.)** $F(x, p) = 0$
  - **Optimizers** $\min_u f(x,u)\ s.t.\ F(x,u) = 0, u \geq 0$
  - **Linear solvers** $Ax = b$
  - **Eigensolvers** $Ax = lBx$
- **Software integration**
- **Performance optimization**



Scope for TOPS

- **Design and implementation of "solvers"**
  - **Time integrators (w/ sens. anal.)** $f(\dot{x}, x, t, p) = 0$
  - **Nonlinear solvers (w/ sens. anal.)** $F(x, p) = 0$
  - **Optimizers** $\min_u f(x,u)\ s.t.\ F(x,u) = 0, u \geq 0$
  - **Linear solvers** $Ax = b$
  - **Eigensolvers** $Ax = lBx$

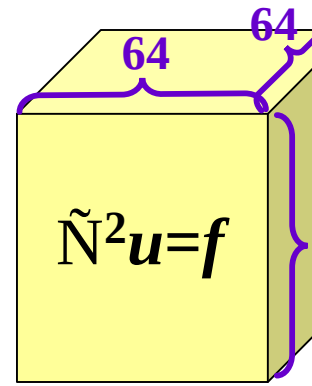


- **Software integration**
- **Performance optimization**

The power of optimal algorithms

- **Advances in algorithmic efficiency rival advances in hardware architecture**
- **Consider Poisson's equation on a cube of size $N=n^3$**

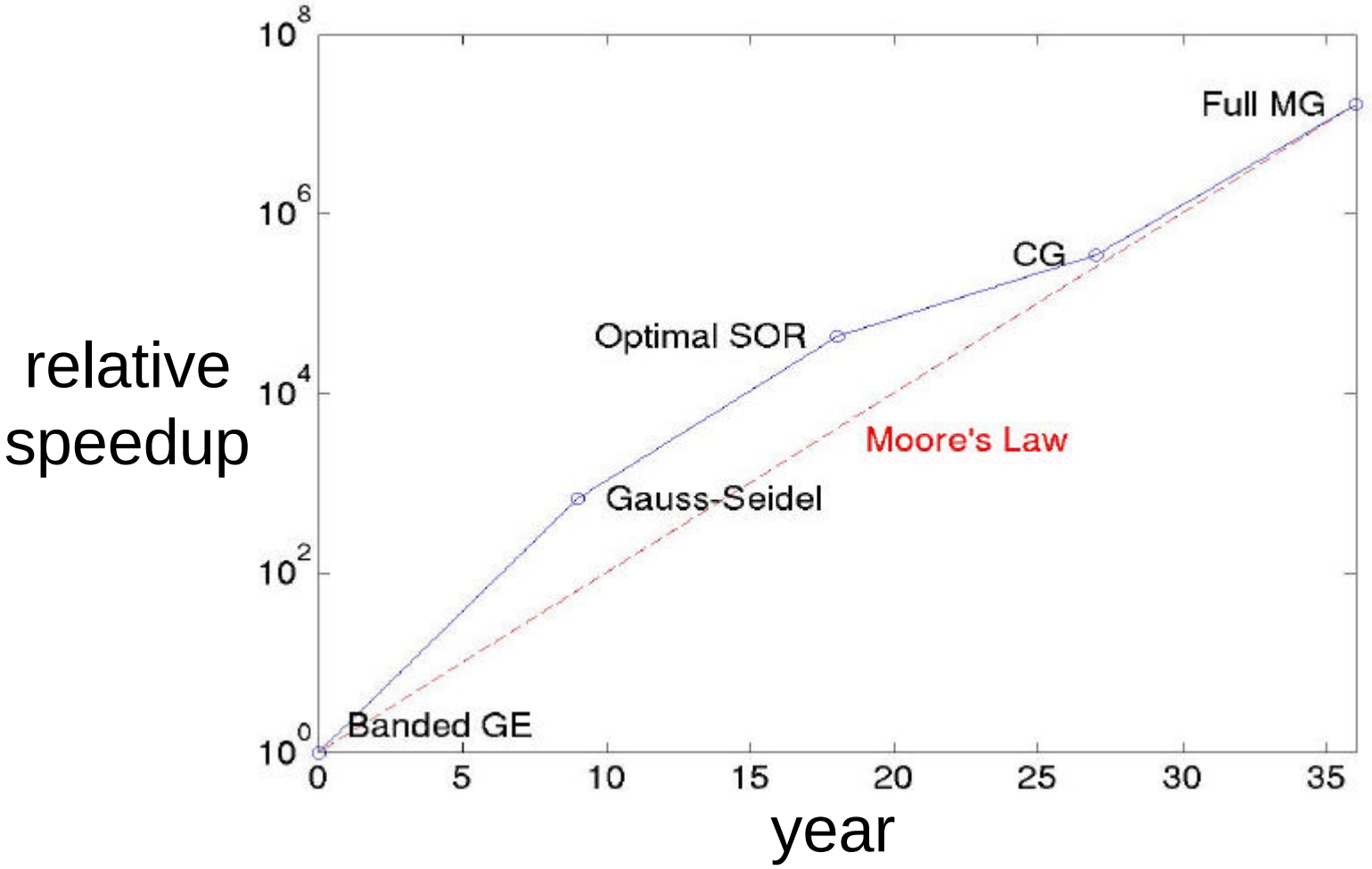
Year	*Method*	*Reference*	*Storage*	*Flops*
1947	GE (banded)	Von Neumann & Goldstine	n^5	n^7
1950	Optimal SOR	Young	n^3	$n^4 \log n$
1971	CG	Reid	n^3	$n^{3.5} \log n$
1984	Full MG	Brandt	n^3	n^3



- **If n=64, this implies an overall reduction in flops of ~16 million***

Algorithms and Moore's Law

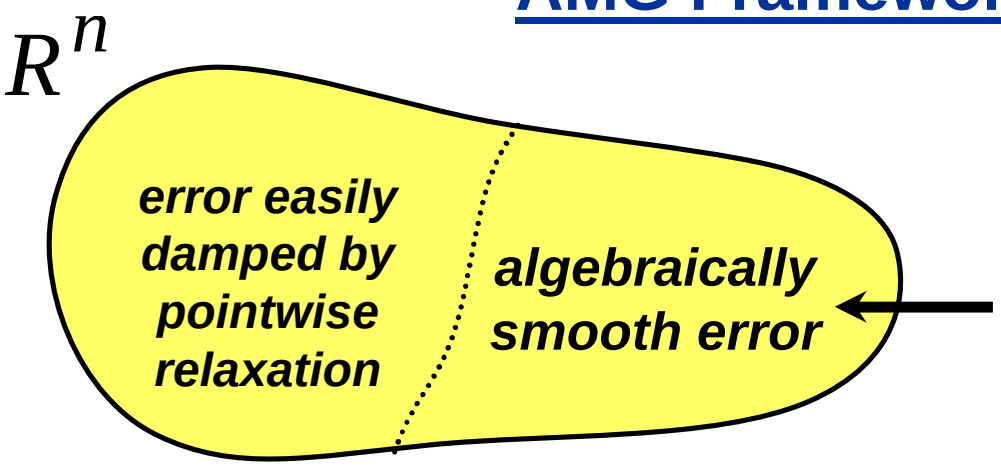
- **This advance took place over a span of about 36 years, or 24 doubling times for Moore's Law**
- **2^{24}»16 million Þ the same as the factor from algorithms alone!**



Where to go past *O(N)* ?

- **Since *O(N)* is already optimal, there is nowhere further "upward" to go in efficiency, but one must extend optimality "outward", to more general problems**
- **Hence, for instance, algebraic multigrid (AMG), obtaining *O(N)* in *indefinite*, *anisotropic*, or *inhomogeneous* problems**

AMG Framework

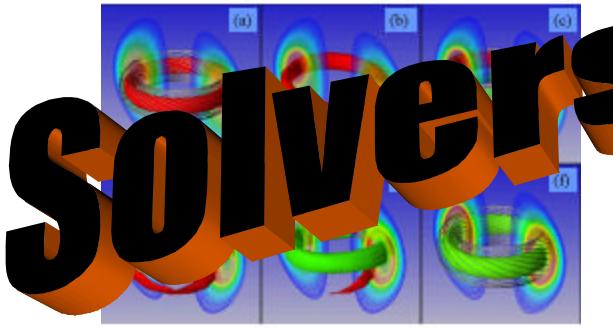


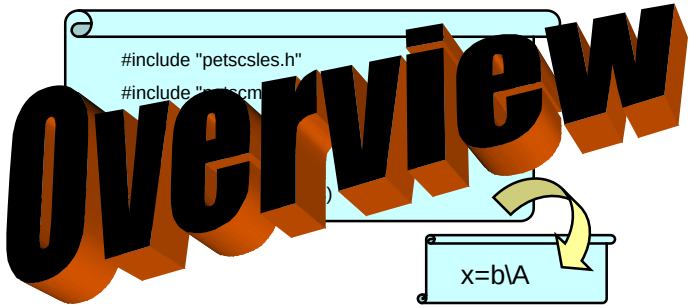
Choose coarse grids, transfer operators, and smoothers to eliminate these "bad" components within a smaller dimensional space, and recur

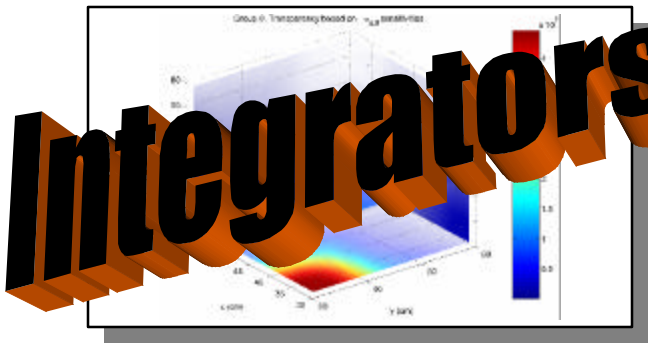
Progress and prospects

- **Progress**
  - **highlighted in five 2-pagers on**
    - **project overview**
    - **scalable solvers**
    - **eigensolvers**
    - **optimizers**
    - **performance**
  - **http://www.osti.gov/scidac/updates2003.html**
- **Prospects**
  - **highlighted in five posters (5pm today)**
- **Balance of talk contains pointers and introductions**

Poster iconography

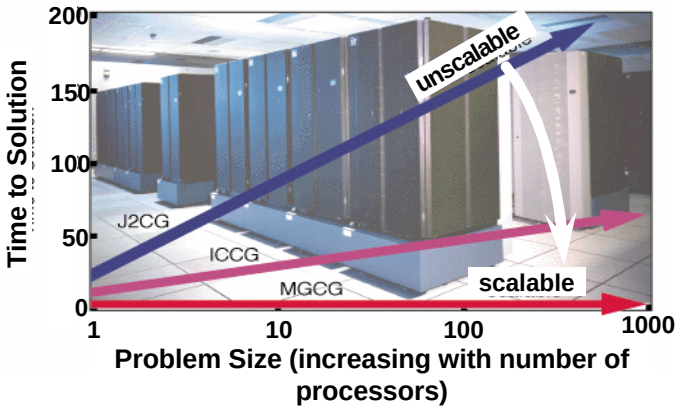


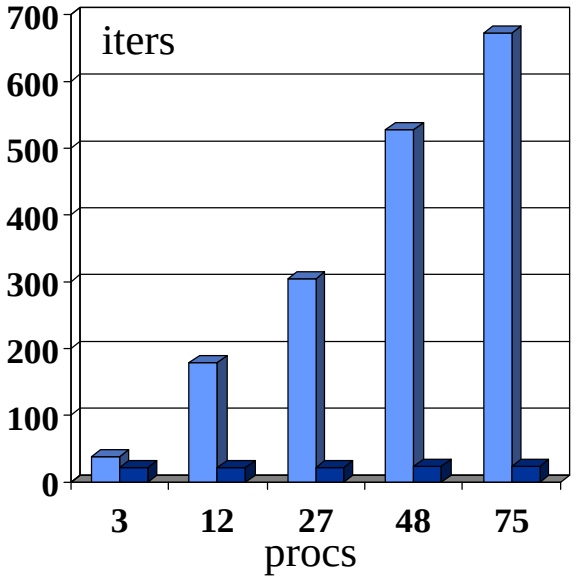


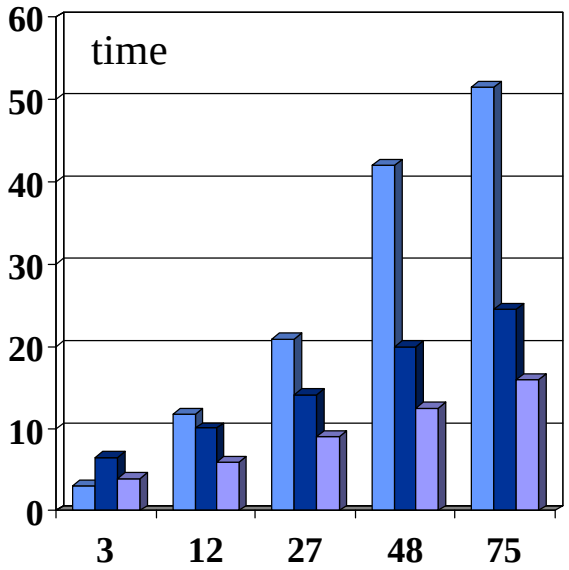


Optimal solvers

- **Convergence rate nearly independent of discretization parameters**
  - **Multilevel schemes for linear and nonlinear problems**
  - **Newton-like schemes for quadratic convergence of nonlinear problems**







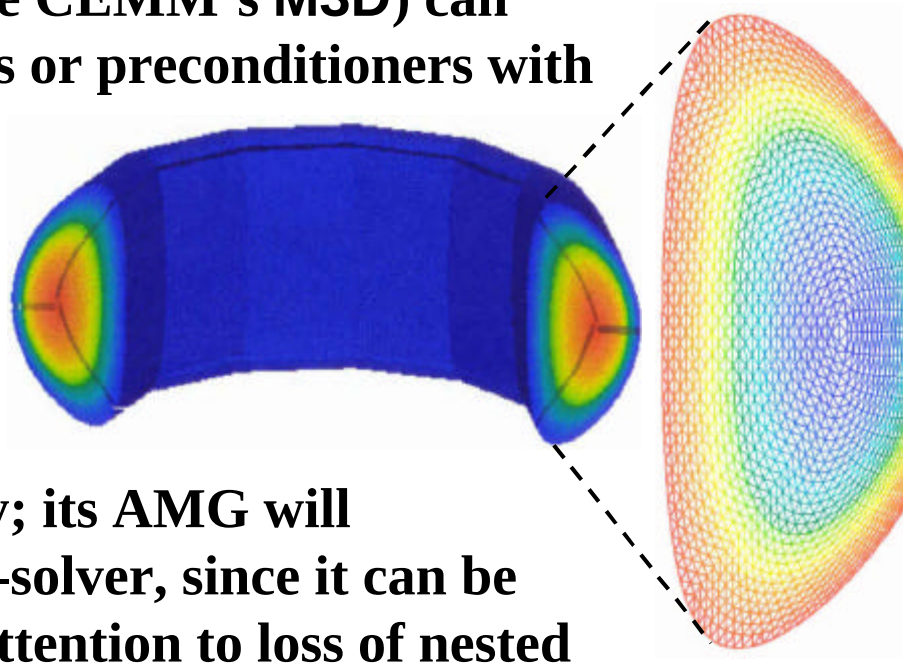
ASM-GMRES
AMG-FMGRES

AMG shows perfect iteration scaling, above in contrast to ASM, bu still needs performance work to achieve temporal scaling, belov on CEMM fusion code M3D, though time is halved (or better) for large runs (all runs: 4K dofs per processor)

ASM-GMRES
AMG-FMGRES
AMG inner

Solver interoperability

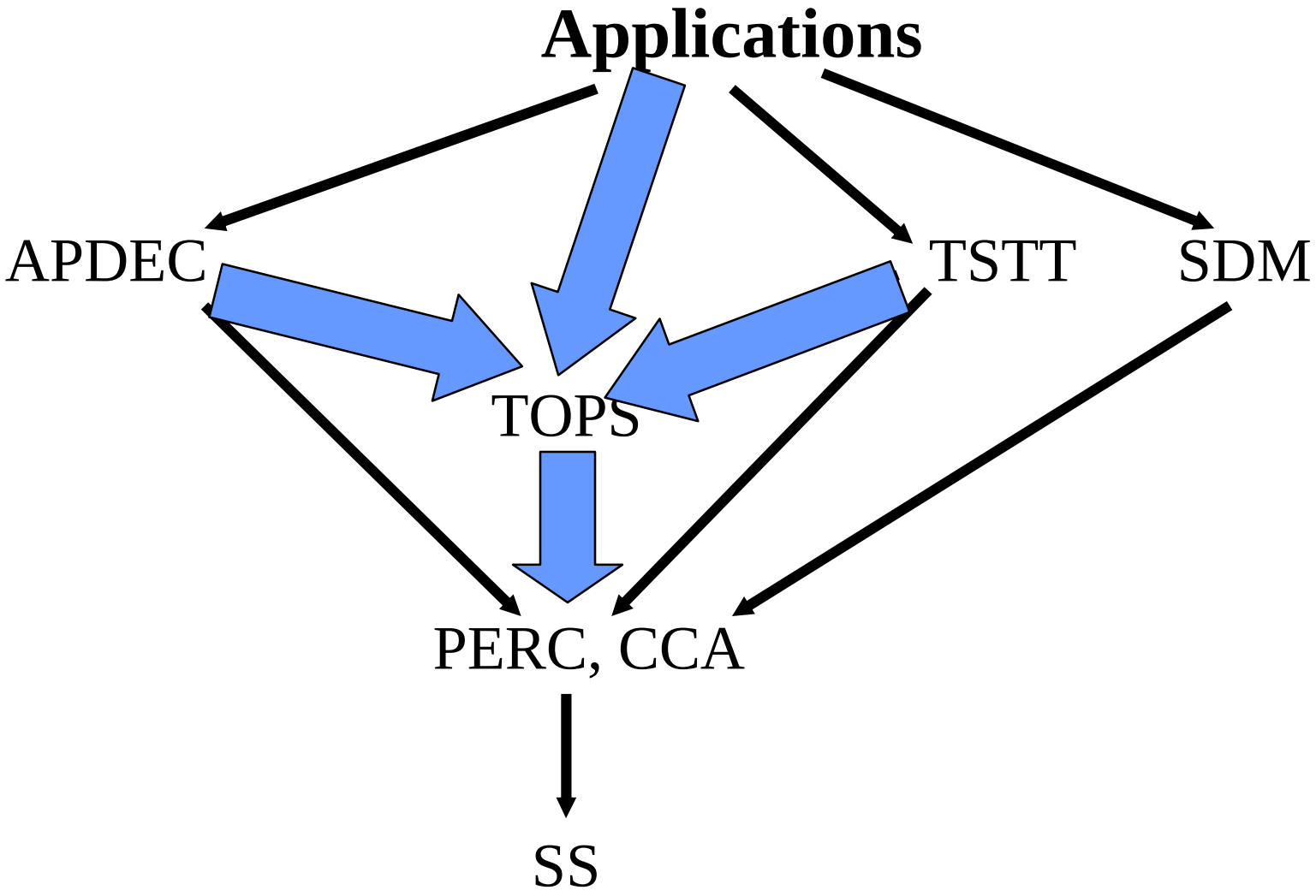
- **Hypre in PETSc**
  - codes with PETSc interface (like CEMM's M3D) can invoke Hypre routines as solvers or preconditioners with command-line switch
- **SuperLU_DIST in PETSc**
  - as above, with SuperLU_DIST
- **Hypre in Chombo**
  - so far, Hypre is level-solver only; its AMG will ultimately be useful as a bottom-solver, since it can be coarsened indefinitely without attention to loss of nested geometric structure; also FAC is being developed for AMR uses, like Chombo
- **Hypre and PETSc both being "SIDL'ized"**
  - one of TOPS' three foci of interaction with CCTTSS



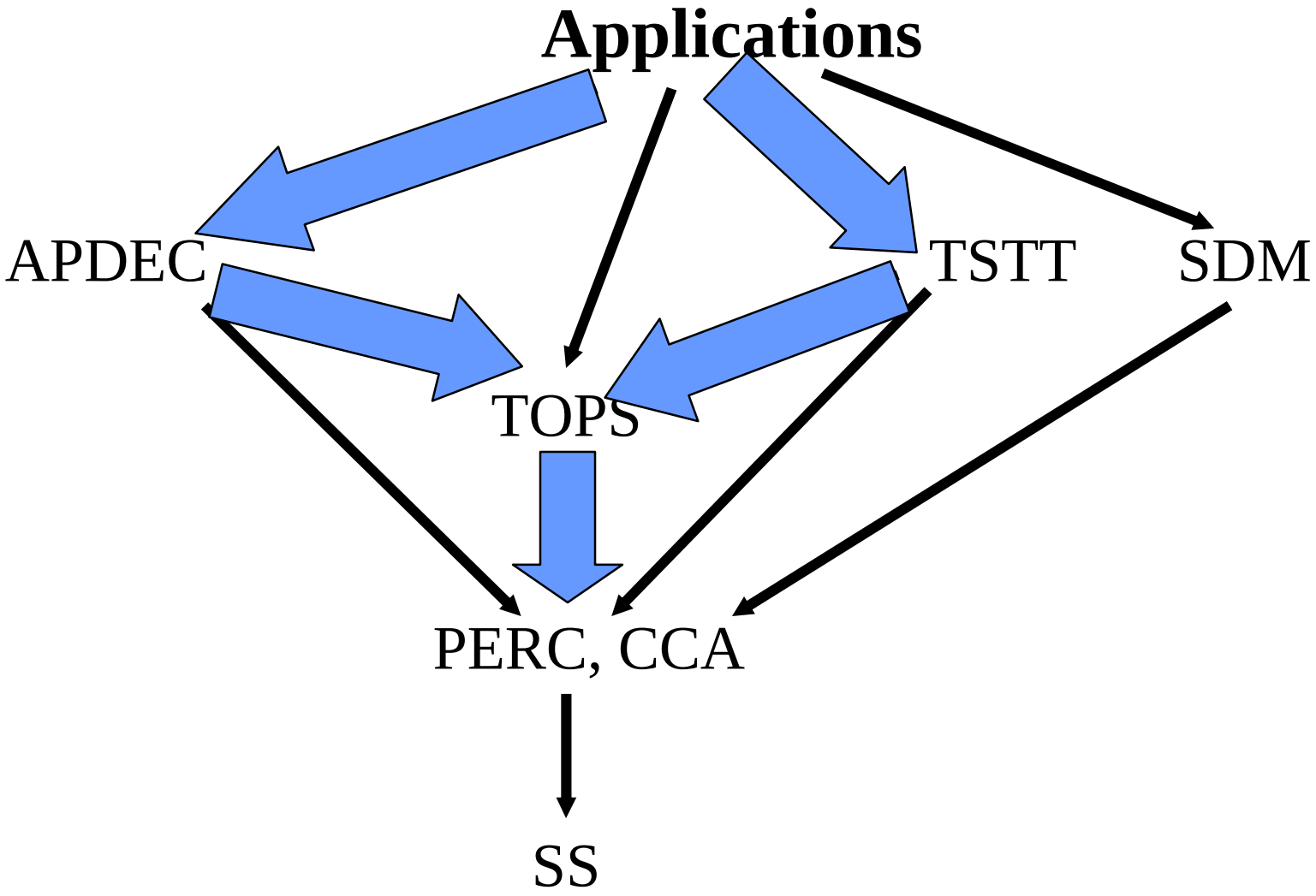
Other solver efforts

- **Have implemented CMRS's model problem (2D periodic Hall MHD) in PETSc and included it in PETSc 2.1.5 release (`ex29.c`)**
  - **Permits order-of-magnitude increase in timestep beyond Courant stability limit for original CMRS code on uniform grid without loss of accuracy in functionals of interest; importance will grow for AMR applications**
- **Will support future 2D and 3D versions of TSI's BOLTZTRAN**
- **Will support matrix-free Newton-Krylov solver for implicit solves on composite AMR grids (APDEC)**
- **Will support preconditioning with economical low-order operators of TSTT's discretizations of high order**

Interaction pathways, 2003*



Interaction pathways, 2005*

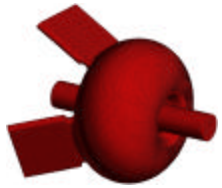


Eigensolvers

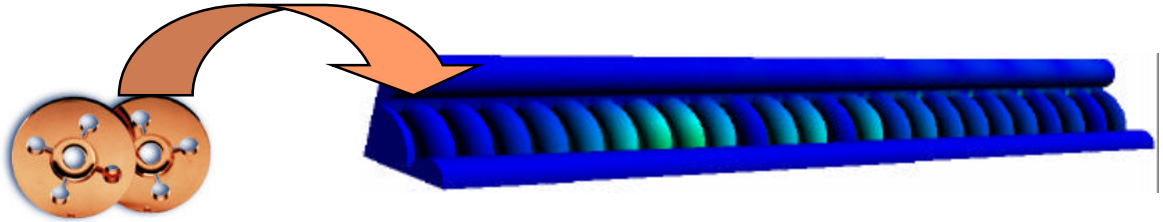
- **No universal eigensolver**
  - **type?**
    - **dense or sparse**
    - **Hermitian or non-Hermitian**
    - **standard ($Ax = \lambda x$) or general ($Ax = \lambda Bx$)**
  - **seek?**
    - **all, extremal, or interior parts of the spectrum**
    - **just eigenvalue *counts* within a spectral range**
    - **eigenvalues themselves, or eigenvalues and eigenvectors together**
  - **resources?**
    - **high or low storage available**
- **With AST, TOPS is pushing the envelope on**
  - **sparse, generalized real symmetric case**
  - **for a cluster of low modes**
  - **under both low storage and high storage conditions**

Eigensolvers for AST

- AST's **Omega3P** is using TOPS software to find EM modes of accelerator cavities, currently lossless (lossy to come)



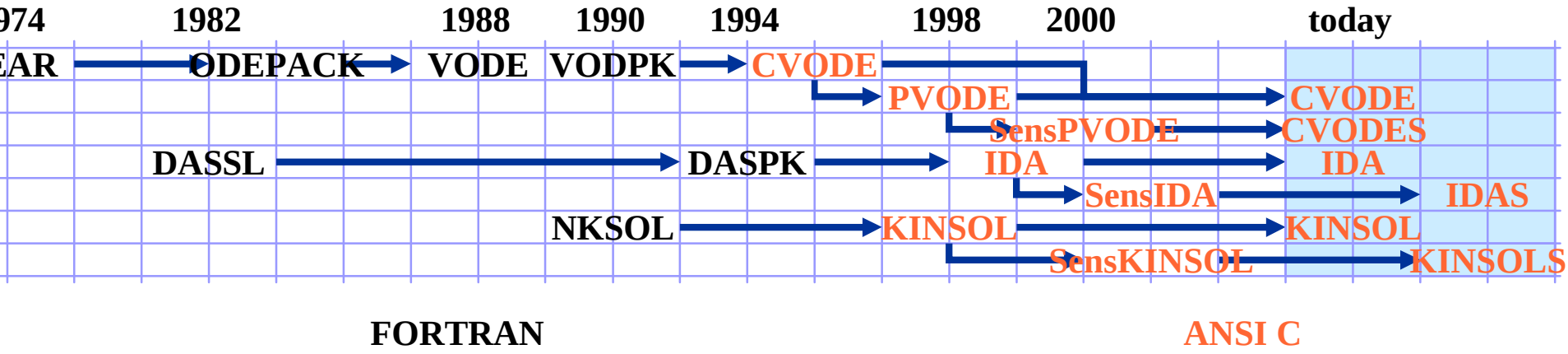
- Methods: Exact Shift-and-Invert Lanczos (ESIL), combining **PARPACK** with **SuperLU** when there is sufficient memory, and Jacobi-Davidson otherwise
- Current high-water marks:
  - 47-cell chamber, finite element discr. of Maxwell's eqs.
  - System dimension 1.3 million
  - 20 million nonzeros in system, 350 million in LU factors
  - *halved analysis time* on 48 processors, scalable to many hundreds



Integrators

SUNDIALS
SUite of Nonlinear and DIfferential/ALgebraic equation Solvers

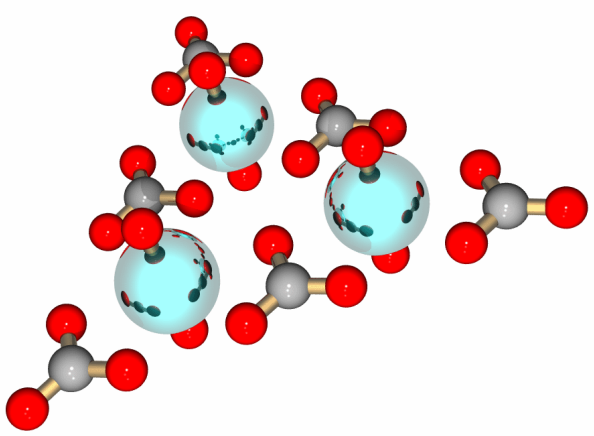
- PVODE, IDA, and KINSOL now wrapped together in SUNDIALS and augmented with forward and adjoint sensitivity analysis capabilities
- Embodies decades of work in variable-order, variable-order method-of-lines and Newton-Krylov solvers at LLNL



- Especially recommended for parameterized applications, requiring uncertainty quantification

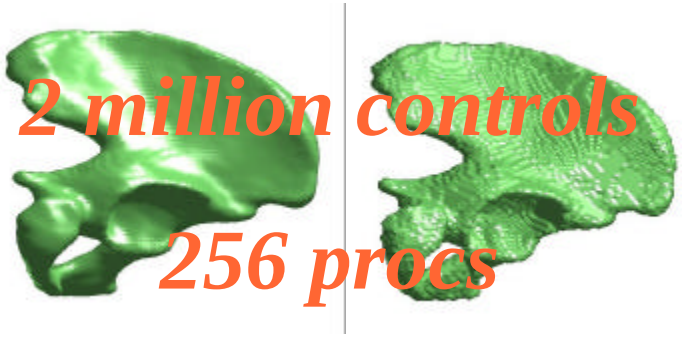
Optimizers

- **Unconstrained or bound-constrained optimization**
  - **TAO (powered by PETSc, interfaced in CCTTSS component framework) used in quantum chemistry energy minimization**



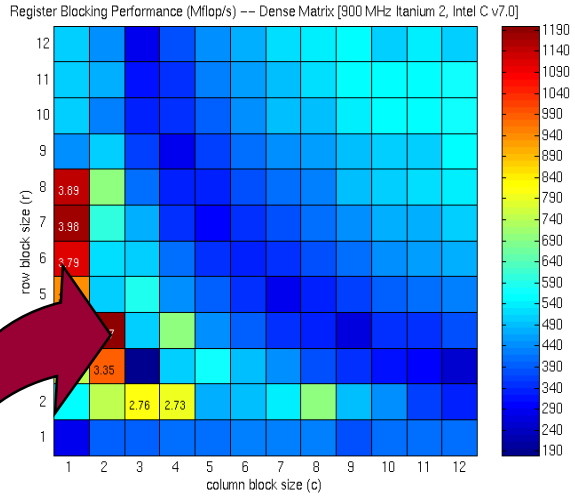
- **PDE-constrained optimization**
  - **Veltisto (powered by PETSC) used in flow control application, to straighten out wingtip vortex by wing surface blowing and sunction**

- **"Best paper" at SC2002 went to TOPS team**
  - **PETSc-powered inverse wave propagation employed to infer hidden geometry**



Performance

- **TOPS is tuning sparse kernels**
  - (Jacobian) matrix-vector multiplication
  - sparse factorization
  - multigrid relaxation
- **Running on dozens of apps/platform combinations**
  - Power3 (NERSC) and Power4 (ORNL)
  - factors of 2 on structured (CMRS) and unstructured (CEMM) fusion apps
- **"Best student paper" at ICS2002 went to TOPS team**
  - theoretical model and experiments on effects of register blocking for sparse mat-vec



Blocking of 4 rows by 2 columns is 4.07 times faster on Itanium2 than default 1´1 blocks

Interoperability/CCA

- Richest interaction so far with any team – fundamental to TOPS
- TOPS helping drive SIDL development
- PETSc, Hypre both being SIDL'ized
- PETSc, TAO part of CCA demos at SC'02
- PLAN: TOPS develop abstract interfaces for linear algebra (including eigenanalysis), nonlinear algebra, unconstrained and constrained optimization
- PLAN: TOPS develop SemiStruct interface for Cartesian AMR codes and composite grid codes

Performance/PERC

- Second richest interaction so far with any other team
- TOPS implicit solver examples providing simple free-standing code targets for PERC
- TOPS application partnerships providing relevant test data to PERC
- PLAN: ORNL acquisition of new Cray X-1 testbed will focus strong interest on PPPL's M3D and PETSc – as sample unstructured implicit app
- PLAN: create insertion path in TOPS production software offerings for Berkeley and UTK successes in performance improvements for sparse kernels

Revelations/Observations

- So far, we sped up ☺ two customer codes (Omega3P, M3D) and *slowed down* ☹ two others (Chombo, CMRS)
  - slowdown experiences are humbling, but extremely beneficial
  - involve "less difficult" base cases, where TOPS is not needed
  - provide a chance for TOPS to provide *one computational physicist's* solution to *another*, through a common solver interface
- Apps groups tend to under-employ complicated iterative libraries on their own
  - underexploitation of available structure
  - underexploitation of algorithmic options
  - underexploitation of profiling tools
- TOPS thinks of its work as *adding options*, *not* making changes
- TOPS can help a lot *before* adding solver options
- TOPS personnel have been learning at least as much as they have been helping – no one is ready to quit yet!

Lessons to date

- **Working with the same code on the same machine vastly speeds collaboration, as opposed to ftp'ing matrices around the country, etc.**
- **Exchanging code templates better than exchanging papers, etc.**
- **Version control systems essential to having any last impact or "insertion path" for solver improvements**
- **"Doing physics" more fun than doing driven cavities**

Questions

Has your solver been unchanged for the past five or ten years?

Is your solver running at 1-10% of machine peak?

Do you spend more time in your solver than in your physics?

Is your discretization or model fidelity limited by the solver?

Is your time stepping limited by stability?

Are you running loops *around* your analysis code?

Do you care how sensitive to parameters your results are?

If the answer to any of these questions is "yes", please tell us at the poster session!

Who to talk with at *this* meeting …

- **Linear solvers**
  (Rob Falgout, Tom Manteuffel, Steve McCormick)
- **Eigensolvers**
  (Esmond Ng)
- **Nonlinear solvers**
  (David Keyes, Carol Woodward)
- **ODE/DAEs/sensitivity**
  (Carol Woodward)
- **Optimizers**
  (Omar Ghattas)
- **Software integration**
  (Rob Falgout, David Keyes)
- **Performance optimization**
  (Jack Dongarra)

Ask about our "0% down" "no payment until 2006" introductory offers on parallel solvers that have won a Bell Prize, a best paper prize, taken ASCI physics apps to 3K processors, and taken chemists to covers of Science and Nature!

Who to talk with at *this* meeting …

- **Linear solvers**
  (Rob Falgout, Tom Manteuffel, Steve McCormick)
- **Eigensolvers**
  (Esmond Ng)
- **Nonlinear solvers**
  (David Keyes, Carol Woodward)
- **ODE/DAEs/sensitivity**
  (Carol Woodward)
- **Optimizers**
  (Omar Ghattas)
- **Software integration**
  (Rob Falgout, David Keyes)
- **Performance optimization**
  (Jack Dongarra)

Falgout, LLNL

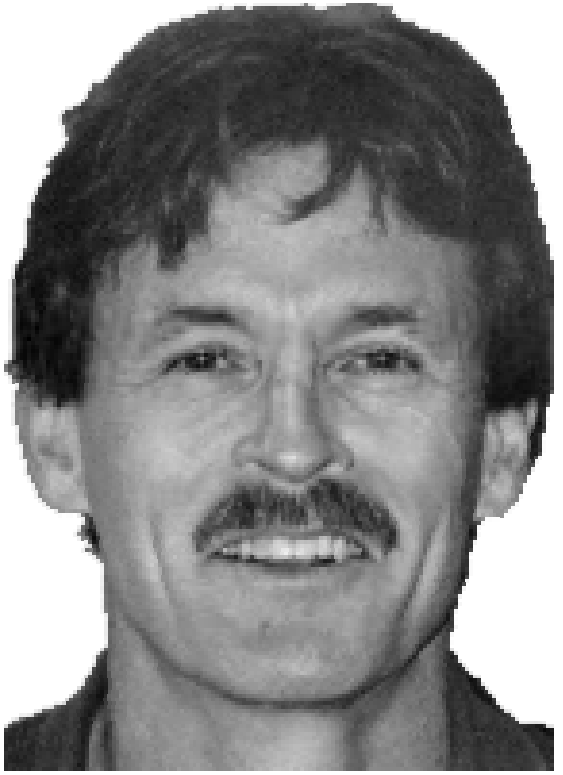
Who to talk with at *this* meeting …

- **Linear solvers**
  (Rob Falgout, Tom Manteuffel, Steve McCormick)
- **Eigensolvers**
  (Esmond Ng)
- **Nonlinear solvers**
  (David Keyes, Carol Woodward)
- **ODE/DAEs/sensitivity**
  (Carol Woodward)
- **Optimizers**
  (Omar Ghattas)
- **Software integration**
  (Rob Falgout, David Keyes)
- **Performance optimization**
  (Jack Dongarra)

Manteuffel, CU

Who to talk with at *this* meeting …

- **Linear solvers**
  (Rob Falgout, Tom Manteuffel, Steve McCormick)
- **Eigensolvers**
  (Esmond Ng)
- **Nonlinear solvers**
  (David Keyes, Carol Woodward)
- **ODE/DAEs/sensitivity**
  (Carol Woodward)
- **Optimizers**
  (Omar Ghattas)
- **Software integration**
  (Rob Falgout, David Keyes)
- **Performance optimization**
  (Jack Dongarra)



McCormick, CU

Who to talk with at *this* meeting …

- **Linear solvers**
  (Rob Falgout, Tom Manteuffel, Steve McCormick)
- **Eigensolvers**
  (Esmond Ng)
- **Nonlinear solvers**
  (David Keyes, Carol Woodward)
- **ODE/DAEs/sensitivity**
  (Carol Woodward)
- **Optimizers**
  (Omar Ghattas)
- **Software integration**
  (Rob Falgout, David Keyes)
- **Performance optimization**
  (Jack Dongarra)

Ng, LBNL

Who to talk with at *this* meeting …

- **Linear solvers**
  (Rob Falgout, Tom Manteuffel, Steve McCormick)
- **Eigensolvers**
  (Esmond Ng)
- **Nonlinear solvers**
  (David Keyes, Carol Woodward)
- **ODE/DAEs/sensitivity**
  (Carol Woodward)
- **Optimizers**
  (Omar Ghattas)
- **Software integration**
  (Rob Falgout, David Keyes)
- **Performance optimization**
  (Jack Dongarra)

Keyes, ODU

Who to talk with at *this* meeting …

- **Linear solvers**
  (Rob Falgout, Tom Manteuffel, Steve McCormick)
- **Eigensolvers**
  (Esmond Ng)
- **Nonlinear solvers**
  (David Keyes, Carol Woodward)
- **ODE/DAEs/sensitivity**
  (Carol Woodward)
- **Optimizers**
  (Omar Ghattas)
- **Software integration**
  (Rob Falgout, David Keyes)
- **Performance optimization**
  (Jack Dongarra)

Woodward, LLNL

Who to talk with at *this* meeting …

- **Linear solvers**
  (Rob Falgout, Tom Manteuffel, Steve McCormick)
- **Eigensolvers**
  (Esmond Ng)
- **Nonlinear solvers**
  (David Keyes, Carol Woodward)
- **ODE/DAEs/sensitivity**
  (Carol Woodward)
- **Optimizers**
  (Omar Ghattas)
- **Software integration**
  (Rob Falgout, David Keyes)
- **Performance optimization**
  (Jack Dongarra)

Ghattas, CMU

Who to talk with at *this* meeting …

- **Linear solvers**
  (Rob Falgout, Tom Manteuffel, Steve McCormick)
- **Eigensolvers**
  (Esmond Ng)
- **Nonlinear solvers**
  (David Keyes, Carol Woodward)
- **ODE/DAEs/sensitivity**
  (Carol Woodward)
- **Optimizers**
  (Omar Ghattas)
- **Software integration**
  (Rob Falgout, David Keyes)
- **Performance optimization**
  (Jack Dongarra)

Dongarra, UT

What we believe

- Many of us came to work on solvers through interests in applications
- What we believe about …
  - applications
  - users
  - solvers
  - legacy codes
  - software

  … will impact how comfortable you are collaborating with us
- So please give us your comments on the next five slides!

What we believe about *apps*

- **Solution of a system of PDEs is rarely a goal in itself**
  - **Actual goal is characterization of a response surface or a design or control strategy**
  - **Solving the PDE is just one forward map in this process**
  - **Together with analysis, sensitivities and stability are often desired**

⇒ **Software tools for PDE solution should also support related follow-on desires**

- **No general purpose PDE solver can anticipate all needs**
  - **Why we have *national laboratories*, not *numerical libraries* for PDEs today**
  - **A PDE solver improves with user interaction**
  - **Pace of algorithmic development is very rapid**

⇒ **Extensibility is important**

What we believe about *users*

- **Solvers are used by people of varying numerical backgrounds**
  - **Some expect MATLAB-like defaults**
  - **Others want to control everything, e.g., even varying the type of smoother and number of smoothings on different levels of a multigrid algorithm**

$\Rightarrow$ **Multilayered software design is important**

- **Users' demand for resolution is virtually insatiable**
  - **Relieving resolution requirements with modeling (e.g., turbulence closures, homogenization) only defers the demand for resolution to the next level**
  - **Validating such models requires high resolution**

$\Rightarrow$ **Processor scalability and algorithmic scalability (optimality) are critical**

What we believe about *legacy code*

- **Porting to a scalable framework does not mean starting from scratch**
  - **High-value meshing and physics routines in original languages can be substantially preserved**
  - **Partitioning, reordering and mapping onto distributed data structures (that we may provide) adds code but little runtime**

⇒ **Distributions should include code samples exemplifying "separation of concerns"**

- **Legacy solvers may be limiting resolution, accuracy, and generality of modeling overall**
  - **Replacing the solver may "solve" several other issues**
  - **However, pieces of the legacy solver may have value as part of a preconditioner**

⇒ **Solver toolkits should include "shells" for callbacks to high value legacy routines**

What we believe about *solvers*

- **Solvers are employed as part of a larger code**
  - **Solver library is not only library to be linked**
  - **Solvers may be called in multiple, nested places**
  - **Solvers typically make callbacks**
  - **Solvers should be swappable**

⇒ **Solver threads must not interfere with other component threads, including other active instances of themselves**

- **Solvers are employed in many ways over the life cycle of an applications code**
  - **During development and upgrading, robustness (of the solver) and verbose diagnostics are important**
  - **During production, solvers are streamlined for performance**

⇒ **Tunability is important**

What we believe about *software*

- **A continuous operator may appear in a discrete code in many different instances**
  - **Optimal algorithms tend to be hierarchical and nested iterative**
  - **Processor-scalable algorithms tend to be domain-decomposed and concurrent iterative**
  - **Majority of progress towards desired highly resolved, high fidelity result occurs through cost-effective low resolution, low fidelity parallel efficient stages**

⇒ **Operator abstractions and recurrence are important**

- **Hardware changes many times over the life cycle of a software package**
  - **Processors, memory, and networks evolve annually**
  - **Machines are replaced every 3-5 years at major DOE centers**
  - **Codes persist for decades**

⇒ **Portability is critical**

Goals/Success Metrics

TOPS users —

- Understand range of algorithmic options and their tradeoffs (*e.g.*, memory *vs.* time, inner iteration work *vs.* outer)
- Can try all reasonable options easily without recoding or extensive recompilation
- Know how their solvers are performing
- Spend more time in their physics than in their solvers
- Are intelligently driving solver research, and publishing joint papers with TOPS researchers
- Can simulate *truly new physics*, as solver limits are steadily pushed back (finer meshes, complex coupling, etc.)

Expectations TOPS has of users

- **Tell us if you think our assumptions above are incorrect or incomplete**
- **Be willing to experiment with novel algorithmic choices – optimality is *rarely* achieved beyond model problems without interplay between physics and algorithmics!**
- **Adopt flexible, extensible programming styles in which algorithmic and data structures are not hardwired**
- **Be willing to let us play with the real code you care about, but be willing, as well to abstract out relevant compact tests**
- **Be willing to make concrete requests, to understand that requests must be prioritized, and to work with us in addressing the high priority requests**
- **If possible, *profile* before seeking help**

TOPS may be for you!

For more information ...

http://www.tops-scidac.org